International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 04 | Apr 2021 www.irjet.net p-ISSN: 2395-0072

Modern Social Media Application on Serverless Architecture with Microservices Pattern

Ankita Dalvi[1], Megha Darda[2], Anshika Kumar[3], Navneet Khanna[4], Vidya Waykule[5]

[1,2,3,4]Undergraduate Students, Department of Computer Engineering, AISSMS COE, Pune, Maharashtra, India
[5]Professor, Department of Computer Engineering, AISSMS COE, Pune, Maharashtra, India

---

Abstract - *Social Media Applications have grown dramatically over the last decade, generating massive ever-growing data exerting an extensive load on the infrastructure. Thus, to efficiently handle such a large quantity of data without the need for managing the underlying infrastructure which has limited capacity and is susceptible to downtime, and also, seamlessly auto-scaling the system and reducing overall costs, serverless architectures come to the rescue.*

The serverless application operates as a collection of interconnected services that involves communication logic between the data and services of the app. Microservices and Serverless Architecture have changed the way we perceive web-based applications. Thus, by using the benefits of Serverless which deals with using the whole Backend as a service (BAAS) and also adding the Microservices code pattern for independence and isolation, we create an architecture that uses the best of both worlds.

Key Words: Serverless Architecture, Microservices, Cloud Computing, BaaS, Social Media Application, Cloud Native.

1. Introduction

In the past decade, social media has become an important part of our everyday life, changing the way we communicate, collaborate, gather information, exchange thoughts and ideas and consequently perceive the world around us.

As of January 2021, 4.2 billion of the world's population are now active on social media, which means around 53% of the world population now uses social media actively as per the annual US 2021 Digital Trends report by Hootsuite, and We are Social.

Such massive usage of the social media platforms generates tremendous amounts of data and a need for a strongly designed and managed infrastructure for which cloud platforms seem to be the best choice.

1.1 Modern Cloud-Native Application

Application development approaches have been constantly changing. Cloud-Native Applications have become popular choices to embrace modern approaches like serverless, microservices, and containers. In Cloud-Native architectures, we can quickly code, build, deploy and manage without compromising security or quality.

Cloud-Native architectures take complete advantage of the latest and best technologies around distributed systems. They are specifically designed to utilize the versatility, reliability, and scalability benefits of the cloud. Cloud-Native Application ensures flexibility of the infrastructure and dependencies, enhances automation capabilities, and improves utilization of the resources.

1.2 Microservices Code Pattern

As Social Media Applications have many functionalities, with each experiencing heavy traffic, Microservices have become the core of cloud-native social media application architecture. Cloud-native applications are collections of microservices that are designed and developed independently and are packaged as lightweight containers and developed into the cloud and typically rely on asynchronous communication to share information with each other.

Individual microservices expose different APIs, each with its own application logic built to execute a single business function. Some microservices handle functionalities that the end-client doesn't directly interface with. When one service fails in a microservice pattern, for whatever reason, other microservices will continue functioning without an issue. Containers make deployment into any cloud platform (AWS, Google, etc.) simpler and can ease the process to quickly scale-in or scale-out. In this manner, the infrastructure utilization becomes fully optimized.

2. Serverless Application

Serverless Applications comprise of third-party Backend-as-a-Service (BaaS) services, and/or that include containers which run custom code on a Functions-as-a-Service (FaaS) platform that is event-triggered where the Cloud Provider dynamically manages the allocation and provisioning of servers. By leveraging serverless, applications draw out the need for a traditional always-on server component. Serverless architectures benefit significantly from easier monitoring, lowered operational cost, complexity, and engineering lead time. AWS Lambda, Google Cloud Functions, Azure Functions, IBM OpenWhisk etc are popular cloud provider choices for going Serverless

2.1. Key Features of Serverless Architecture

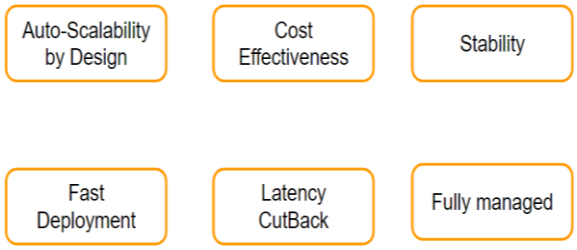

Fig -1: Key features of Serverless architectures

3. Serverless Microservices Architecture

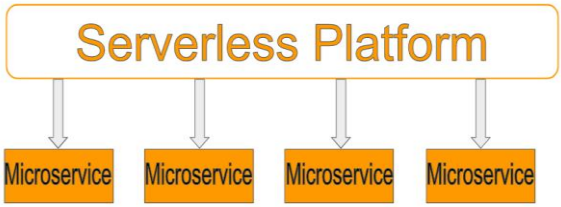

Fig -2: Microservices Serverless Architecture

In the Microservices Pattern, each job or functionality is isolated separately. The serverless platform by Cloud Provider invokes appropriate service on an event trigger. Each of these services can also be broken down into smaller functions. These functions and services can access a file system, a database, or use third-party services. It is also important to use best code practices, as one could end up with too many functions which are harder to manage and can result in a lot of cognitive overhead.

```
service: serverless-social-network
provider: aws
functions:
  usersCreate:
    handler: handlers.usersCreate
    events:
      - http: post users/create
  commentsCreate:
    handler: handlers.commentsCreate
    events:
      - http: post comments/create
```

Fig -3: Microservices code pattern in Serverless

(Refer fig 3) Event-driven serverless architectures change how we approach application development, its management, and its architecture. Within a microservices approach, the intent is to be responsive to the interface and that is the primary mechanism for interaction with the logic whereas within a serverless approach, the intent is to be responsive to events that occur like an HTTPinvocation, and an API is only a mechanism for generating events.

For a data-heavy and traffic-heavy platform like social media, the serverless platform with microservices code pattern if implemented correctly could also make it simpler to move to a pure microservices architecture in the future, which has its own benefits if the traffic is known to be consistent with predictable growth and also provides control over the infrastructure of the system.

3.1. Code and Data Isolation between Microservices

We can deploy multiple microservices each having full isolation of code. Furthermore, each microservice can have multiple versions deployed simultaneously. Though mostly isolated, microservices share some resources like Databases, cache, etc. While this sharing has certain advantages, a microservices-based application needs to maintain code- and data-isolation between microservices. To mitigate unwanted sharing, one must architect the system carefully. If these patterns of isolation are not reliable one can formally enforce separation by using multiple projects for the same application. One could also opt for a hybrid approach of project and service isolation.

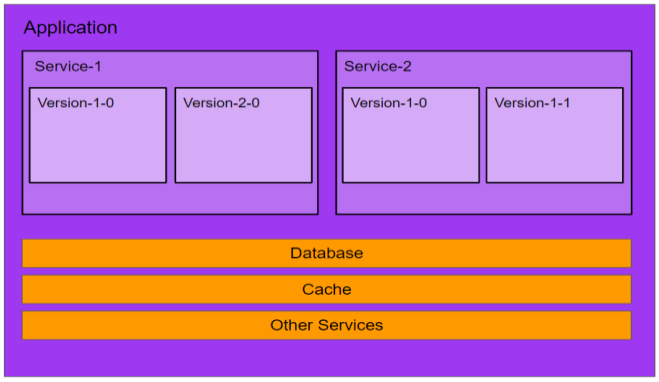

Fig -4: Isolation in Serverless Compute Service

3.2. Tradeoffs of using Serverless Microservices

- Reduced overall control over the infrastructure
- Increased reliance on vendor dependencies demands more for a third-party provider's trust
- Security risk
- Comparatively immature technology, unclear best practices
- Unless architected correctly, the architectural complexities could provide a poor user experience
- Lack of operational tools.

4. Proposed Architecture

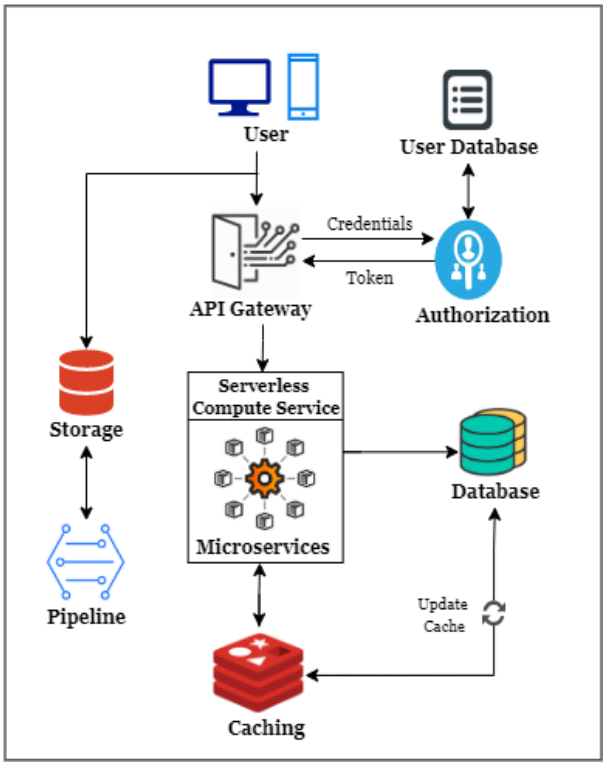

Fig -5: Architecture of Social Media Application on Serverless Microservices

Every time the user uses the social media application through their browser or app they send a request. This request could be anything like creating comments, getting feed, editing profiles, etc. Each of these requests would be a call to some API. API calls are accepted by the API gateway and then it aggregates the various services required to fulfill them and returns the appropriate result.

Firstly, we get and verify the authentication credentials from the user and return a response to the client. We also use the generated token by the authorization service to verify the identity of users in the backend services. Event-driven architectures are pretty popular in modern web application development and exhibit more reliable behavior in a distributed environment. These events invoke the appropriate microservice which is deployed inside the Serverless Compute Service. The key to the functioning of these microservices inside the Serverless Microservice Architecture would be the unidirectionality of code.

The microservice first fetches the cache and if the cache entry is found empty then the request accesses the database. Then the cache is updated for future use. Unstructured Data from Social Media Applications make NoSQL databases a primary choice.These platforms being Media-heavy need Data Pipelines to compress the media to save storage space. One could use a messaging service that is triggered when a media is uploaded into the storage and then data pipelines work on that uploaded media.

5. CONCLUSIONS

With serverless being a buzzword, if the architecture is implemented correctly, Microservices inside Serverless for a Social Media Application would significantly take off the load to manage and maintain the infrastructure and foster innovation, and also, ensure a seamless experience for the users.

REFERENCES

[1] Kim M., Lee H. (2011) SMCC: Social Media Cloud Computing Model for Developing SNS Based on Social Media. In: Lee G., Howard D., Ślezak D. (eds) Convergence and Hybrid Information Technology. ICHIT 2011. Communications in Computer and Information Science, vol 206. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-24106-2_34

[2] C.Veni, "A Study on Social Networks and Cloud Computing.", International Journal of Engineering Trends and Applications (IJETA) – Volume 5 Issue 1, Jan-Feb 2018

[3] R. A. P. Rajan, "Serverless Architecture - A Revolution in Cloud Computing," 2018 Tenth International Conference on Advanced Computing (ICoAC), Chennai, India, 2018, pp. 88-93, doi: 10.1109/ICoAC44903.2018.8939081.

[4] Rathore, Nitin & Rajavat, Anand & Patel, Margi. (2020). Investigations of Microservices Architecture in Edge Computing Environment. 10.1007/978-981-15-2071-6_7.

[5] Shafiei, Hossein & Khonsari, Ahmad & Mousavi, Payam. (2019). Serverless Computing: A Survey of Opportunities, Challenges and Applications. 10.13140/RG.2.2.32882.25286.

BIOGRAPHIES

ANKITA DALVI,
Computer Engineering Student at AISSMS COE,
Savitribai Phule Pune University

MEGHA DARDA,
Computer Engineering Student at AISSMS COE,
Savitribai Phule Pune University

ANSHIKA KUMAR,
Computer Engineering Student at AISSMS COE,
Savitribai Phule Pune University

NAVNEET KHANNA,
Computer Engineering Student at AISSMS COE,
Savitribai Phule Pune University

VIDYA WAYKULE,
Professor, Department of Computer Engineering,
AISSMS COE, Pune-01.